(ENVIRONMENTAL MONITORING PHASE 5)

What is IoT environmental monitoring?

IoT environmental monitoring is a process that uses Internet of Things (IoT) technology to collect data about the environment, such as air quality, temperature, and humidity levels. This data can then be analysed to better understand the indoor and outdoor environment and make informed decisions about how to reduce the impact of negative aspects of the local environment on the business. Alternatively, it can be used to change business activities to help protect the planet or the local community.
These IoT-based systems can be used to detect issues in the environment that are largely invisible, normalised or taken for granted. Allowing businesses to take action by reducing their negative environmental footprint and protecting employees, visitors and the community at large.
IoT is, therefore, becoming increasingly popular across a range of industries for its ability to provide statistics, real-time data and insights that can help businesses understand their environmental impact and how to reduce it, as well as comply with environmental protection standards and policies.

What are the benefits of using IoT-based environmental monitoring?

There are a number of benefits associated with using an IoT-based environmental monitoring system, including:

- **Improved understanding of the environment via data**: With real-time data feeds being supplied by remotely deployed IoT sensors, businesses and organisations can better understand and quantify the environment. From here, targeted actions can be taken to reduce environmental impact or to spot problems, such as excessive CO2, noise or airborne chemicals as they occur.
- **Improved efficiency**: With real-time data, organisations can identify and address any problems long before they become more serious. By employing warning alarms, businesses can be more reactive and proactive. This can result in a better working environment, cost savings and less downtime.

What are the four basic steps of environmental monitoring?

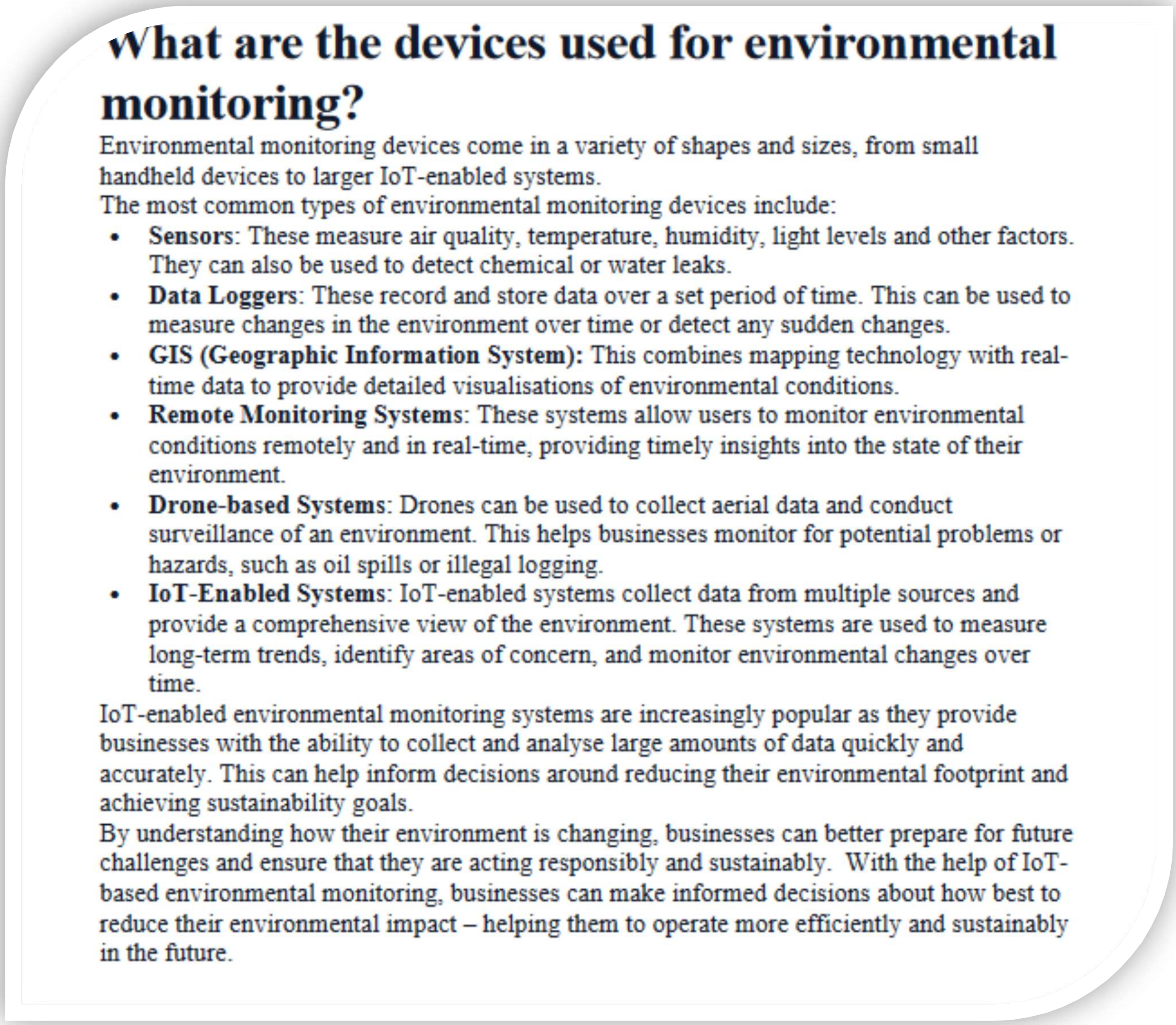
There are four critical components for IoT-based environmental monitoring to support vital insights and decision-making:

1) Observation (Monitor the Environment and Collect Data):

The first step in the environmental monitoring process is to observe and collect data. This involves using sensors or other IoT devices to measure factors such as air quality, temperature, and humidity levels.
These connected IoT devices gather data about the environment and transmit it to a central hub. From here, the data can be reviewed in real-time or used for further analysis off line. Often these systems produce unexpected results and temporal variances. For example, high CO2 levels when offices are highly populated could explain drowsiness or loss of concentration. This can also apply to public spaces such as bars and restaurants where invisible environmental factors may be making the consumer experience uncomfortable.

2) Analysis (Measure Data):

The next step is to analyse the data collected by IoT devices. This includes looking at trends over time, identifying areas of concern, and any correlations between environmental variables, time of day, behaviours and the relationships between indoor and outdoor metrics. IoT sensing devices pick out key points of the data that indicate everything from chemical and water leaks to air pollution levels. This data analysis can help businesses measure their environmental footprint and make informed decisions about how to reduce their environmental impact.



What are the devices used for environmental monitoring?

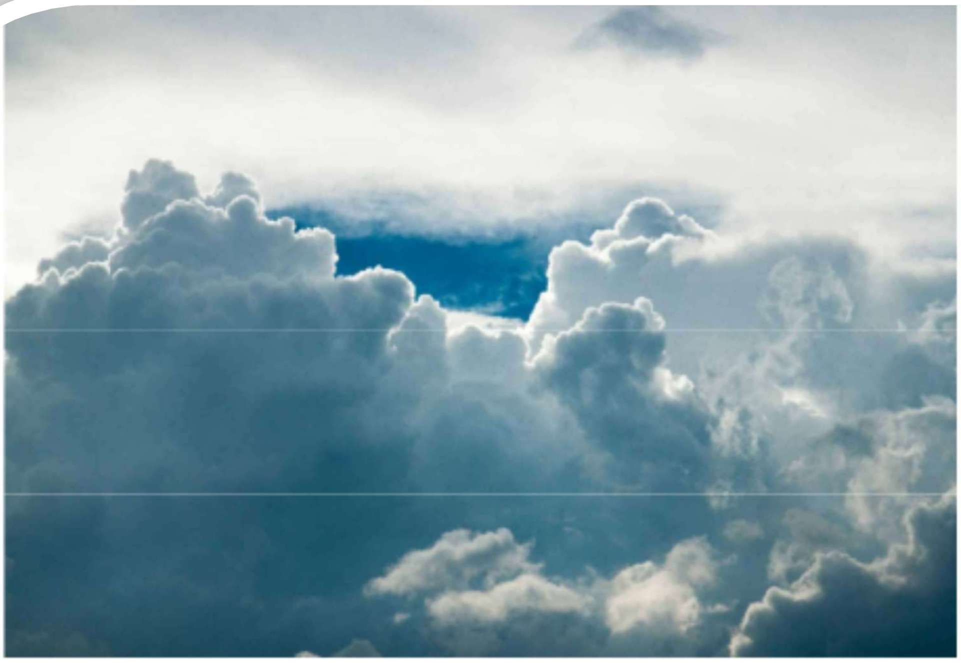
Environmental monitoring devices come in a variety of shapes and sizes, from small handheld devices to larger IoT-enabled systems.

The most common types of environmental monitoring devices include:

- **Sensors**: These measure air quality, temperature, humidity, light levels and other factors. They can also be used to detect chemical or water leaks.
- **Data Loggers**: These record and store data over a set period of time. This can be used to measure changes in the environment over time or detect any sudden changes.
- **GIS (Geographic Information System):** This combines mapping technology with real-time data to provide detailed visualisations of environmental conditions.
- **Remote Monitoring Systems**: These systems allow users to monitor environmental conditions remotely and in real-time, providing timely insights into the state of their environment.
- **Drone-based Systems**: Drones can be used to collect aerial data and conduct surveillance of an environment. This helps businesses monitor for potential problems or hazards, such as oil spills or illegal logging.
- **IoT-Enabled Systems**: IoT-enabled systems collect data from multiple sources and provide a comprehensive view of the environment. These systems are used to measure long-term trends, identify areas of concern, and monitor environmental changes over time.

IoT-enabled environmental monitoring systems are increasingly popular as they provide businesses with the ability to collect and analyse large amounts of data quickly and accurately. This can help inform decisions around reducing their environmental footprint and achieving sustainability goals.

By understanding how their environment is changing, businesses can better prepare for future challenges and ensure that they are acting responsibly and sustainably. With the help of IoT-based environmental monitoring, businesses can make informed decisions about how best to reduce their environmental impact – helping them to operate more efficiently and sustainably in the future.



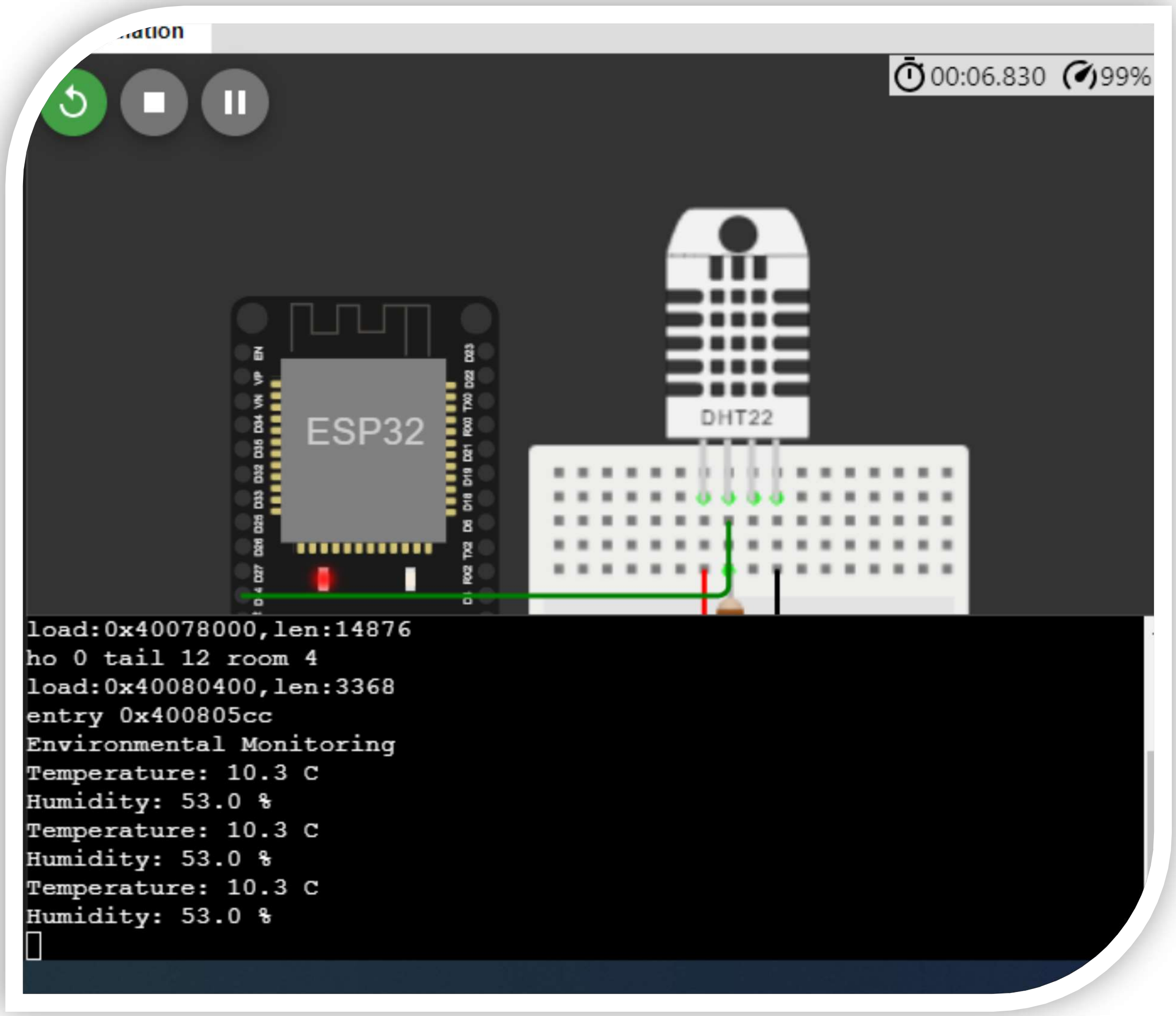
1) Air-quality monitoring:

Industrial processes, like burning fuel, emit air pollutants and organic compounds that can negatively impact human health and the environment.
Whether it's from industrial processes, car exhausts or herds of cattle, the carbon monoxide, hydrocarbons and greenhouse gases emitted must be monitored to ensure good air quality and to protect the wider environment.
Indoor spaces are also subject to pollutants. For example, man-made fibres and materials emit volatile organic compounds (VOCs) over time which are detrimental to human health.
Excessive dust and airborne particles are not good for respiratory health and can affect those with COPD and contain allergens for those sensitive to them.
The variance between indoor and outdoor air quality is also important. For example, opening a window to alleviate CO2 levels can cause more problems if outdoor air quality pollution is considered more harmful.

Input:

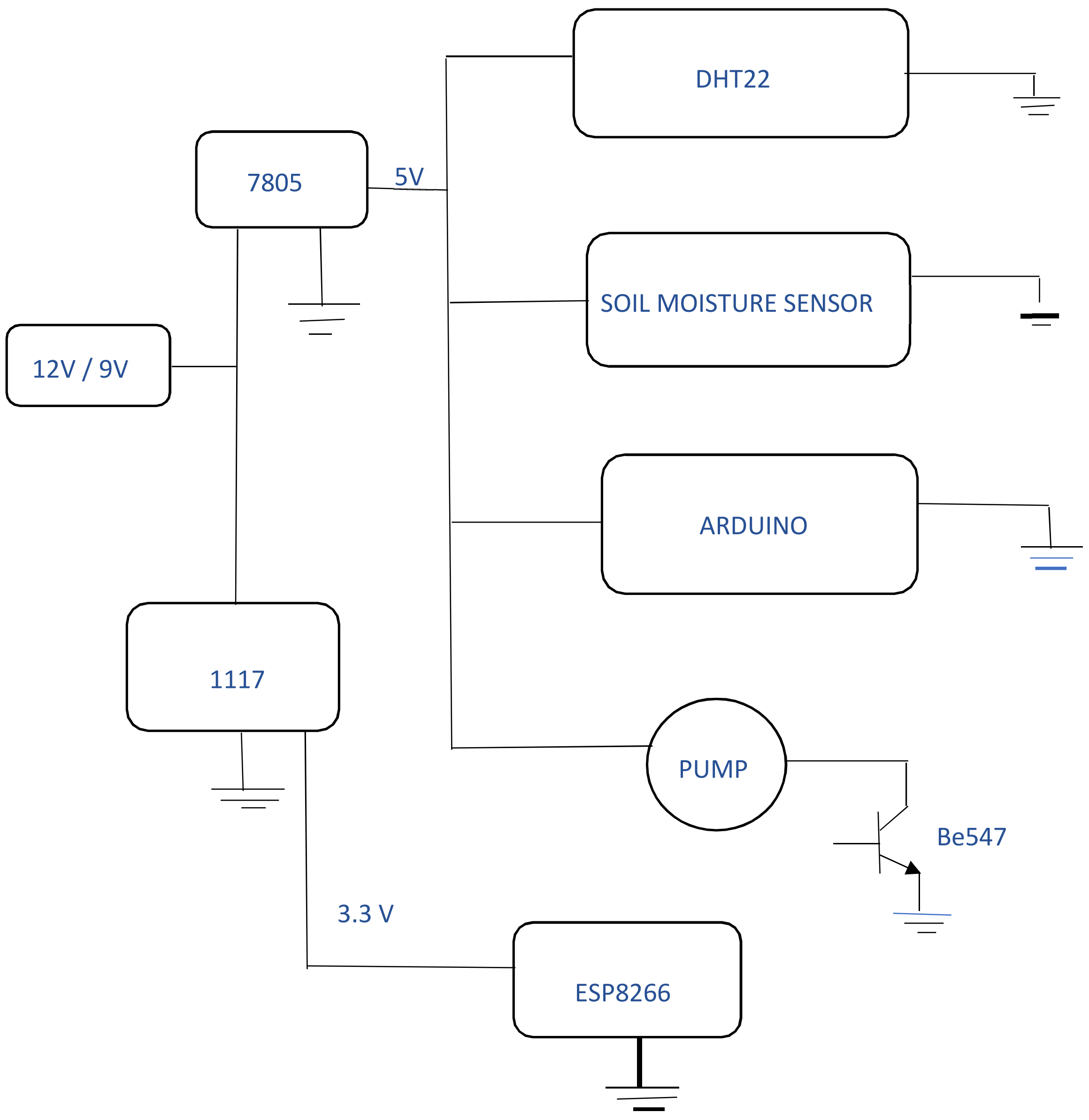
```
print("Environmental Monitoring")

#import BlynkLib
from machine import Pin
from time import sleep
import dht
import time

sensor = dht.DHT22(Pin(14))
#sensor = dht.DHT11(Pin(14))

while True:
    sensor.measure()
    temp = sensor.temperature()
    hum = sensor.humidity()
    #temp_f = temp * (9/5) + 32.0
    print('Temperature: %3.1f C' %temp)
    #print('Temperature: %3.1f F' %temp_f)
    print('Humidity: %3.1f %%' %hum)
    time.sleep(1)
```

Output:



```
load:0x40078000,len:14876
ho 0 tail 12 room 4
load:0x40080400,len:3368
entry 0x400805cc
Environmental Monitoring
Temperature: 10.3 C
Humidity: 53.0 %
Temperature: 10.3 C
Humidity: 53.0 %
Temperature: 10.3 C
Humidity: 53.0 %
```

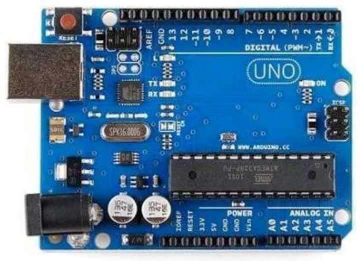
Block Diagram



The farm monitoring system is a mixture of hardware and software additives. The hardware part includes embedded systems and software program is the Arduino ide. The Arduino ide displays readings from sensors are inserted using the hardware. The special sensors used are temperature and humidity sensor and soil moisture sensor. The facts gathered with the aid of the sensors is sent to the Arduino UNO microcontroller. The gathered information may be displayed in an Arduino IDE. An ESP-01 module is hooked up with the Arduino to facilitate notifying service which updates the farmers each 10 seconds approximately the climate conditions of the subject. This project is aided with many hardware. This proposed technology is an amalgamation of different sensors, microcontroller and communication medium to help the farmers to work on their farms.

3.1 Arduino uno

Arduino is a microcontroller to control the working of the sensors and manage the working of the device. The Uno version of Arduino is implemented in this project. It was developed by Arduino CC. The Arduino board comes with various number of pins. The pins are categorized as output and input pins. The input pins accept digital as well as analog pins. It has 14 digital pins and 6 analog pins. It accepts 7 to 20 volts of power for working. It also has an USB port. The Uno was the first version of Arduino to be introduced in the Arduino family.



3.2 Soil Moisture Sensor

The Soil Moisture sensor is used to sense moisture content in the soil. It checks the volume of water content or moisture present in the soil. The calculations are done in the soil moisture sensor through coefficients. It estimates the volume of water content in the soil. It detects the water content in the soil and gets and sends the analog signals which is shown digitally. It transmits the signals containing information or data or values of the condition of soil to Arduino to further process it and display.

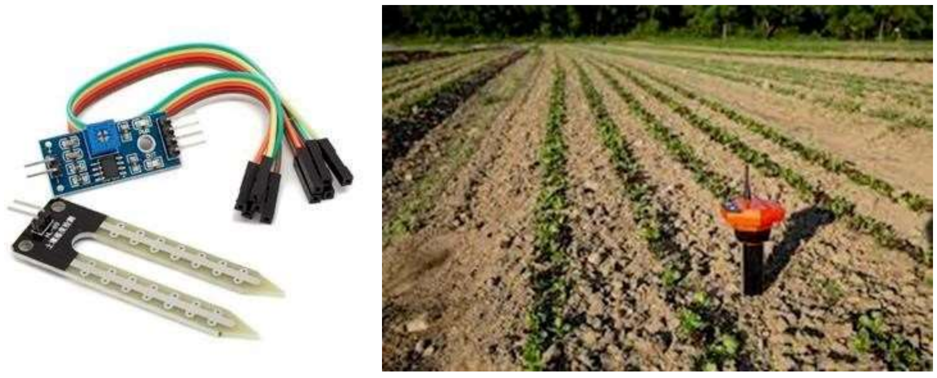


Fig 9. Soil Moisture Sensor

3.3 Nokia 5110 LCD Display Interface with Arduino

This LCD device is mainly used in Arduino but it can be connected with any 3.3V controller. These LCDs are used in Nokia 3110/5110 cell phones. It is a very cheap monochrome LCD module made of 84 x 48 pixels. It can be used to display graphics and text together. This display is based on the PCD8544 driver.

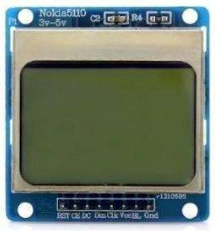


Fig 10. LCD Screen

Pin configuration of this device is almost like the 16x2 LCD module only instead of 8 data pins one serial data in (Din) pin and one clock are there.

The list of the pins and their description are listed below.

RST: Pin type active low, so 0V Resets the LCD

CE: Cheap Enable is used to enable the device before sending anything to the LCD

DC: Data/Command is used to select between Rata Register or Command Register

DIN: Data In is used to send information serially to the display. It could be
Data or Command

CLK: Clock is used to synchronize the display with the controller

VCC: To power, the pin 5V or 3.3V is applied here ***BL***: This

pin is used to power the Backlight of the display

GND: This is used to ground the device.

Things we need

- Nokia 5110 Display
- Arduino
- Resistors
  1. 1k Ohms x 5 Nos.
  2. 330 Ohms / Potentiometer 1k
- Jumper Wires
- Bread Board

Connection Diagram:

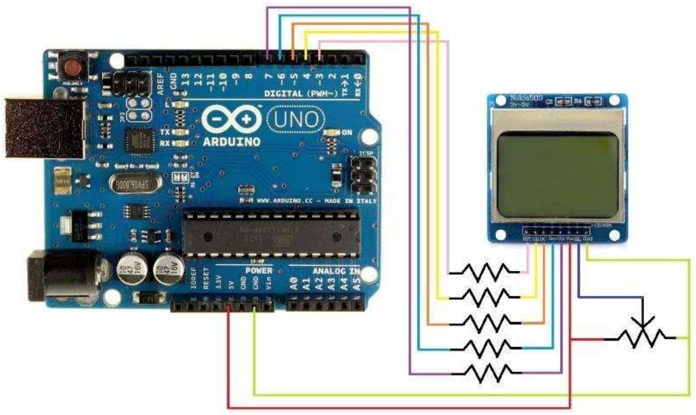
pin 7 - Clock (CLK)

pin 6 - Data In (DIN)

pin 5 - Data/Command select (D/C)

pin 4 - Chip Enable/select (CE/CS)

pin 3 - Reset (RST)



3.4 ESP-01

ESP-01 Module is basically a low-cost esp8266 module, with built-in WIFI. It was created as an Arduino WIFI module but it can also be programmed to work as standalone. Although this module is cheap but working with it is a little difficult. As it is not a breadboard-friendly module it would be a bad choice for a beginner.

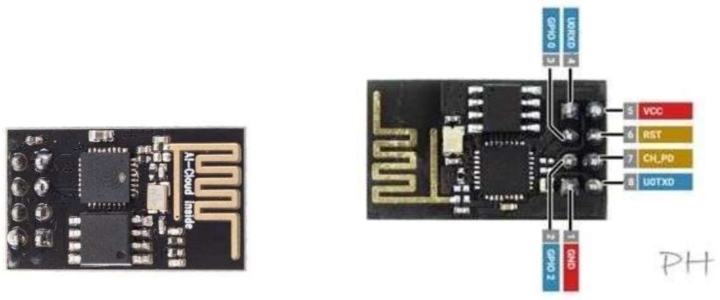


Fig 12. ESP-01

Less I/O pins make it a little difficult to use in standalone mode in projects. It has 8 pins.

- Pin1: Ground Pin
- Pin2: GPIO2 Pin • Pin3: GPIO0 Pin

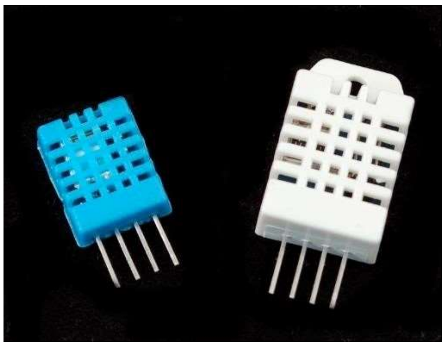
- Pin4: RXD is UART data receive pin.
- Pin5: Vcc is for powering the Module. Only 3.3V power is required.
- Pin6: RST is for external reset. It's active Low in nature.
- Pin7: CH_PD is an active-high pin for Chip enable. • Pin8: TXD is UART data send pin.

This project is about the interfacing of a DHT22 module with Arduino UNO. The data for temperature and humidity are displayed on the Serial Terminal.

Components Required:

1. Arduino Uno
2. DHT22 Module
3. Data Cable
4. Jumper wires

DHT11 and DHT22 sensors are very basic and slow but are great for hobbyists who want to do some basic data logging. The DHT sensors are made of two parts, a capacitive humidity sensor, and a thermistor. There is also a very basic chip inside that does some analog to digital conversion and spits out a digital signal with the temperature and humidity. The digital signal is fairly easy to read using any microcontroller.



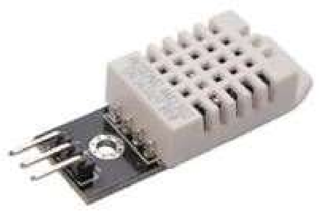


Fig 13. DHT 22

There are two versions of the DHT sensor, they look a bit similar and have the same pinout, but have different characteristics. Here are the specs:

DHT11:

- Ultra-low-cost
- 3 to 5V power and I/O
- 2.5mA max current use during conversion (while requesting data)
- Good for 20-80% humidity readings with 5% accuracy
- Good for 0-50°C temperature readings ±2°C accuracy
- No more than 1 Hz sampling rate (once every second)
- Body size 15.5mm x 12mm x 5.5mm
- 4 pins with 0.1" spacing

DHT22:

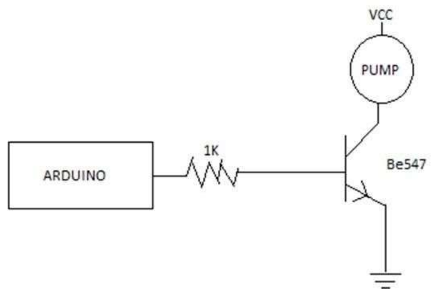
- Low cost
- 3 to 5V power and I/O
- 2.5mA max current use during conversion (while requesting data)
- Good for 0-100% humidity readings with 2-5% accuracy
- Good for -40 to 80°C temperature readings ±0.5°C accuracy
- No more than 0.5 Hz sampling rate (once every 2 seconds)
- Body size 15.1mm x 25mm x 7.7mm
- 4 pins with 0.1" spacing

As you can see, the DHT22 is a little more accurate and good over a slightly larger range. Both use a single digital pin and are 'sluggish' in that you can't query them more than once every second or two.

Pinout: VCC supplies power for the module. You can directly connect it to the 5V pin on the Arduino. Data pin transmits the temperature and humidity data in digital form. GND is the Ground Pin and needs to be connected to the GND pin on the Arduino.

3.6 DC Motor

DC motors are a key component for many agricultural applications, often offering a most efficient form of motion, particularly when solar and battery power is utilised. Some of our own drives are relied on under challenging conditions - for example, they have been in use on Mars for years (although not for agricultural purposes, yet). But DC motors also function in tough agricultural conditions flawlessly and efficiently. Motors, gearheads, sensors, batteries and controllers all constitute the basic building blocks for complex applications. Our own mobile apps include cloud connectivity and give our customers access to a range of functions, including retrieval of current driving data and positions, customisation of parameters and fleet management. All components are verified by our specialists and then perfectly matched to each other. This allows us to offer users a system solution from a single source.



FLOWCHART

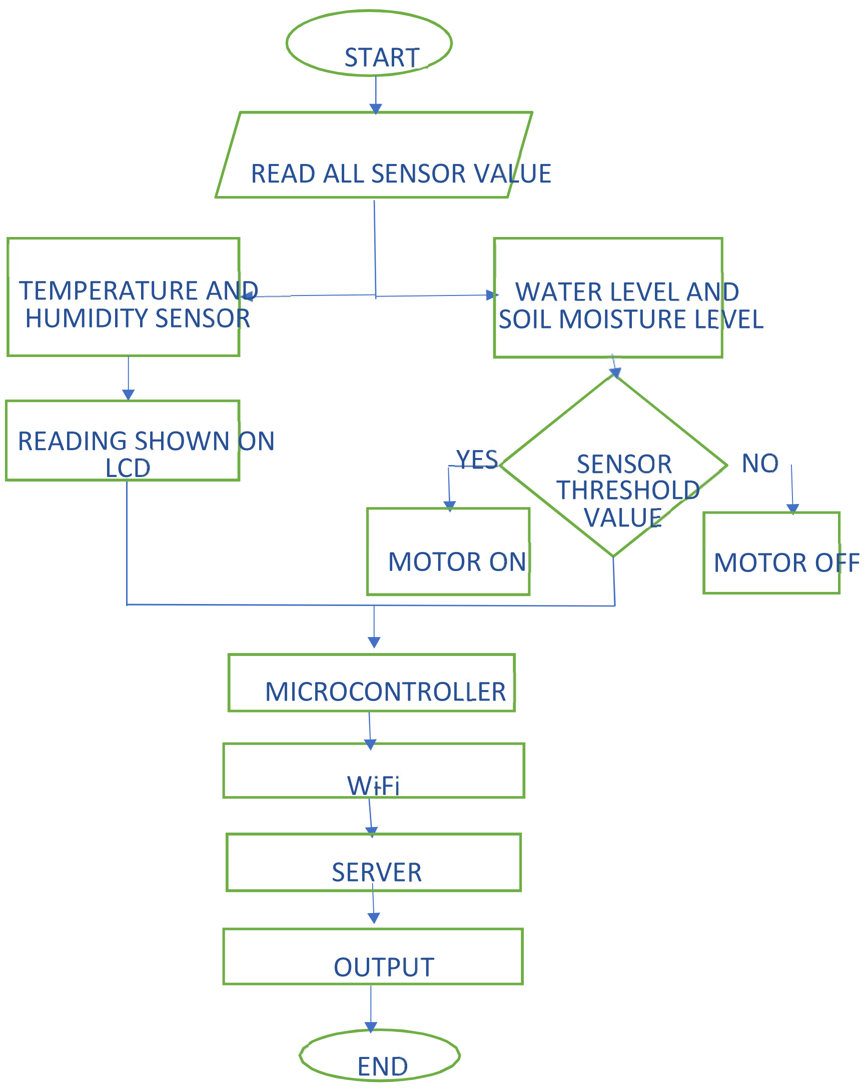


Fig. 15 Flowchart of the project

WORKFLOW

In the power circuit, 9v power comes from the adaptor, and flows to two channels first a 7805 regulator and a 1117 regulator. From the 7805 regulator 5v current passes to DHT22 sensor, soil moisture sensor, Arduino and the pump. While from the 1117 regulator 3.3 v passes to ESP8266 device.

Digital and analog data is fed from the soil moisture sensor and the temperature and humidity sensor to the controller, the controller processes the data and primarily has 3 outputs. temperature readings and moisture levels are displayed on the lcd device while the data is also pushed into ESP8266 device with the help of UART scripting and then it is fed into the cloud, while the controller controls the pump through a npn transistor which is the third from of output.

SOFTWARE PROGRAM

```
#include <SPI.h>

#include <Adafruit_GFX.h>

#include <Adafruit_PCD8544.h>

#include <DHT.h>

float t, h, m1,  m2;

Adafruit_PCD8544 display = Adafruit_PCD8544(6, 5, 4, 3, 2);

#define DHTPIN 12

#define DHTTYPE DHT22

DHT dht (DHTPIN, DHTTYPE);

void setup()

{

  Serial.begin(9600);

  while             (!Serial)

  continue;
  display.begin();
  display.setContrast(35);
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(BLACK);
  display.setCursor(0, 0);
  display.println(" IoT Based");
  display.println("Smart Farming");
  display.display(); delay(2000);
  display.clearDisplay();
  display.display(); pinMode(13,
  OUTPUT); pinMode(11, INPUT);
  digitalWrite(13, LOW); dht.begin();
}
void loop() {
  float h = dht.readHumidity(); float t =
  dht.readTemperature(); float m =
  abs(100-(analogRead(A0)/10.23)); if
  (isnan(h) || isnan(t)) {
    Serial.println("Failed   to   read   from   DHT   sensor!");
    return;
  }
  int it = t*100; int ih =
  h*100; int im = m; bool
  pc = digitalRead(11);
  digitalWrite(13, pc);
  display.clearDisplay();
  display.setCursor(0, 0);

  display.println("Smart

  Farming");

  display.print("Temp: ");

  display.println(t,1);

  display.print("Hum: ");

  display.println(h,1);

  display.print("Soil: ");

  display.println(m,0); if (pc ==

  1) display.print("Pump on");

  display.display();


  // Send Data to Cloud StaticJsonDocument<200>

  doc;

  doc["it"] = it;

  doc["ih"] = ih;

  doc["im"] = im;

  doc["pc"] = pc;

  serializeJson(doc, Serial);

  delay(500);

}
```

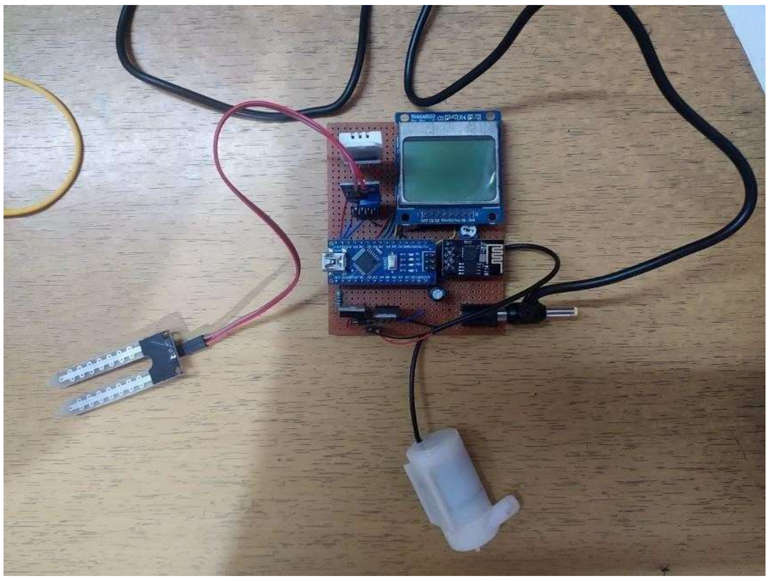
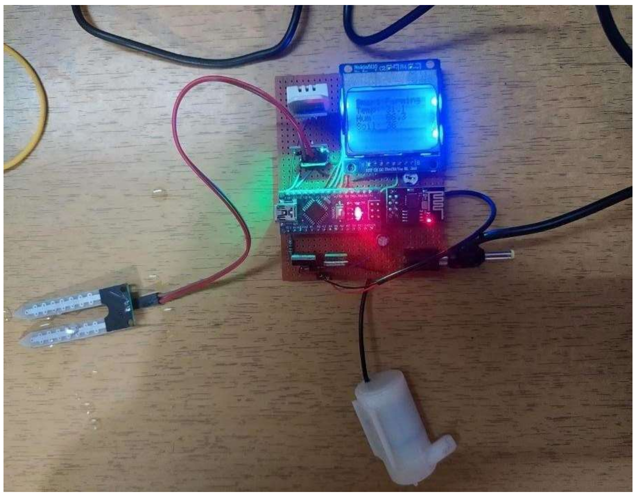
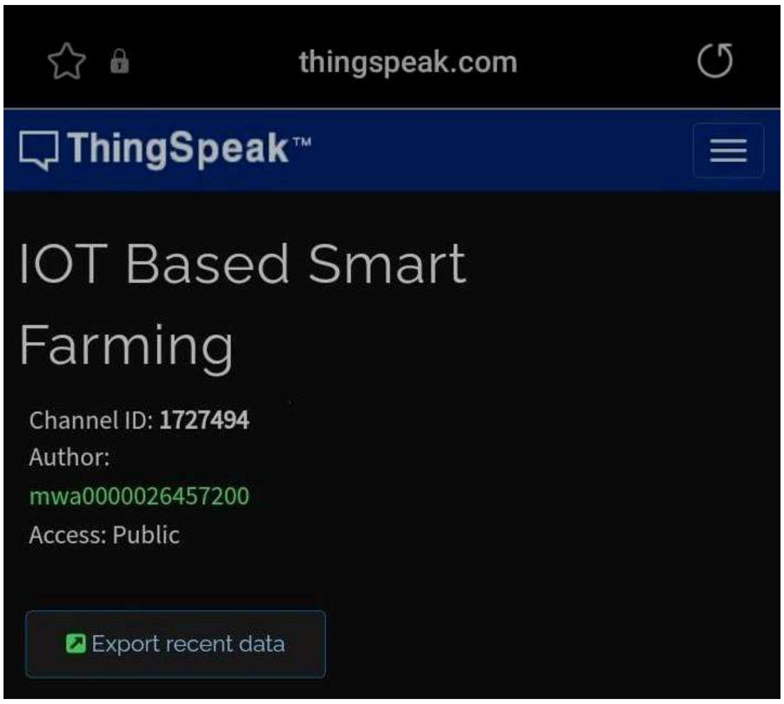


Fig 16. Prototype in off condition



6.1 Basic Operation

The ESP-01 of the prototype at first connects to UART website through Wi-Fi. Here, we are connecting it with the website ThinkSpeak (https://thingspeak.com/pages/learn_more). ThingSpeak™ is an IoT analytics platform service that allows you to aggregate, visualize and analyze live data streams in the cloud. ThingSpeak provides instant visualizations of data posted by your devices to ThingSpeak. With the ability to execute MATLAB® code in ThingSpeak you can perform online analysis and processing of the data as it comes in. ThingSpeak is often used for prototyping and proof of concept IoT systems that require analytics.


thingspeak.com

ThingSpeak™

IOT Based Smart Farming

Channel ID: **1727494**
Author:
mwa0000026457200
Access: Public

Export recent data

Fig. 18 ThinkSpeak provides a user-friendly platform

6.2 Operation 1: Temperature and Humidity Reading

The DHT-22 helps us to get the temperature and humidity value at any particular time. Keeping a check in the temperature and humidity value will help the user in taking care of livestock and special plants. From DHT22, the signal gets passed to ESP8266 which further passes the signal to the website from where the user can get notified even when away from the field.

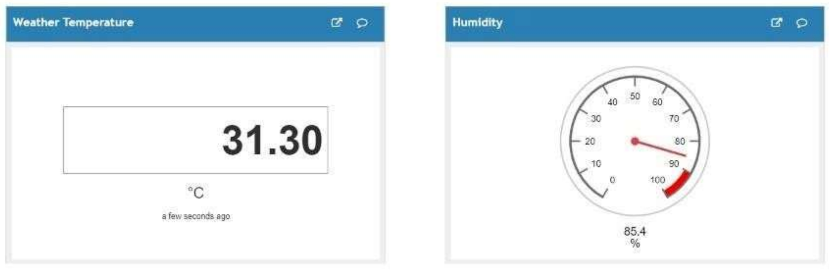


Fig. 19 Record of temperature and humidity

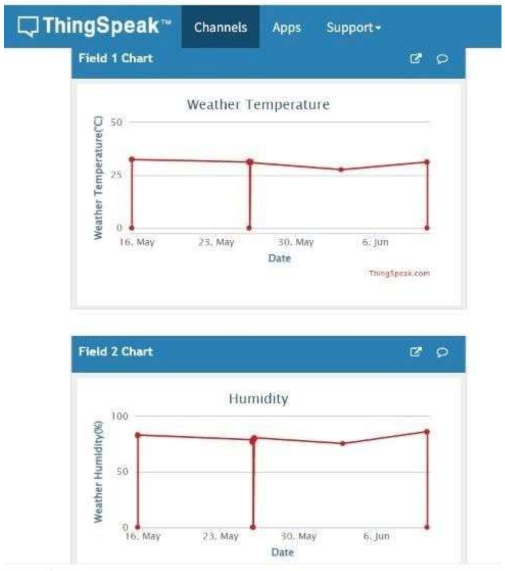


Fig. 20 Temperature and Humidity Graph

6.3 Operation 2: Soil Moisture Determination

The soil moisture sensor determines the moisture content of the soil. If the value is below the threshold value, it sends signal to the DC motor i.e., the pump and automatically the pump gets on and waters the field.

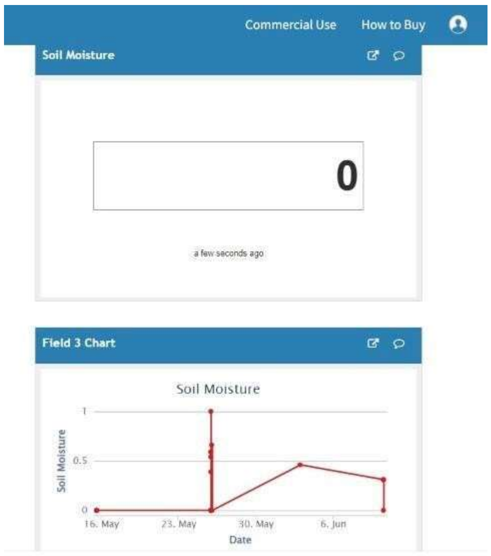
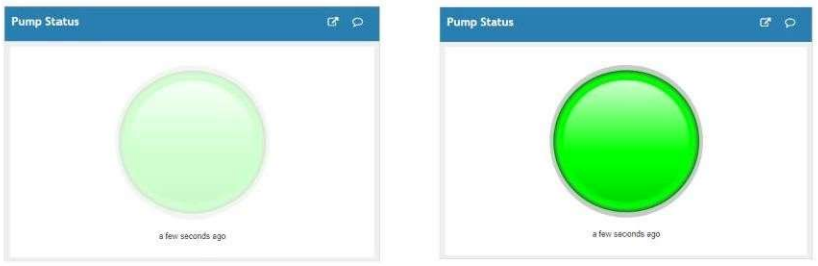


Fig. 21 Moisture content of the field along with the graph


Pump Status
a few seconds ago
Pump Status
a few seconds ago